Application System/400

SX09-1286-00

System/38-Compatible COBOL Reference Summary

IBM

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page ix.

First Edition (June 1994)

This edition applies to the System/38-Compatible COBOL feature of the IBM* ILE* COBOL/400* licensed program, (Program 5763-CB1), Version 3 Release 0 Modification 5, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the proper edition for the level of the product.

Order publications through your IBM representative or the IBM branch serving your locality. Publications are not stocked at the address given below.

A form for readers’ comments is provided at the back of this publication. If the form has been removed, you may address your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East,
North York, Ontario, Canada M3C1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "Communicating your Comments to IBM" for a description of the methods. This page immediately precedes the Readers' Comment Form at the back of this publication.

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1994. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
IBM is a registered trademark of International Business Machines Corporation, Armonk, N.Y.

Contents

© Copyright IBM Corp. 1994

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent product, program, or service that does not infringe on any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, Connecticut, USA 06904-2501.

Changes or additions to the text are indicated by a vertical line (|) to the left of the addition or change.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*), used in this publication, are trademarks of the IBM Corporation in the United States or other countries:

Application System/400
AS/400
COBOL/400
IBM
ILE
Operating System/400
OS/400

© Copyright IBM Corp. 1994

About This Summary

| This publication summarizes the formats of the
| System/38-Compatible COBOL language to be used on the AS/400*
| system.

Who Should Use This Summary

This publication is for programmers familiar with the COBOL language.

When users work with System/38-Compatible COBOL programs on the AS/400 system, they can do so only by accessing an AS/400 feature called the System/38 environment, which emulates the System/38. This summary provides all of the System/38-Compatible COBOL statements you may need to refer to when programming in the System/38 environment.

Before using this summary, you should have a basic understanding of the AS/400 system and the Control Language (CL) applicable to System/38-Compatible COBOL.

What You Should Know

Before you use this summary you should be familiar with the following:

- Card files (such as `READER, PUNCH, and PUNCHPRINT`), card devices, and related language elements are not supported by System/38-Compatible COBOL in the System/38 environment. Programs written with references to card devices and related language elements will be syntax checked but compilation in the System/38 environment will fail. These programs are compatible with the System/38 and compilation of such programs can be performed on a System/38 on which the COBOL Licensed Program (Program 5714-CB1) is installed. A note indicating that card files, card devices and related language elements are not supported in System/38-Compatible COBOL will appear where these references occur in the manual.

© Copyright IBM Corp. 1994

About This Summary

- *System/38-Compatible COBOL User's Guide and Reference*, *SC09-1814*, which more fully documents all information contained in this summary.
- *Programming: Control Language Programmer's Guide*, SC41-8077, which contains the basic concepts of the control program functions.
- You should be familiar with your display station (also known as a work station), and its controls. There are also some elements of its display and certain keys on the keyboard that are standard regardless of which software system is currently running at the display station, or which hardware system the display station is hooked up to. Some of these keys are:
  - Cursor movement keys
  - Command keys
  - Field exit keys
  - Insert and delete keys
  - The Error Reset key.

  This information is contained in the *Communications: Remote Work Station Guide, SC41-0002*.
  - You should know how to operate your display station when it is hooked up to the IBM AS/400 system and running AS/400 software. This means knowing about OS/400 and the Control Language (CL) to do such things as:
  - Sign on and sign off the display station
  - Interact with displays
  - Use Help
  - Enter control commands and procedure commands
  - Call utilities
  - Respond to messages.

  To find out more about this operating system and its control language, refer to:
  - *Programming: Control Language Reference*, SC41-0030
  - *Programming: Control Language Programmer's Guide*, SC41-8077
  - *Programming: Reference Summary, SX41-0028*
  - *System/38 Environment Programmer's Guide and Reference*, SC41-9755

- You should know how to call and use certain utilities available on the AS/400 system:
  - The Screen Design Aid (SDA) utility used to design and code displays. This information is contained in the *Application Development Tools: Screen Design Aid User's Guide and Reference, SC09-1340*.
  - The Source Entry Utility (SEU), which is a full-screen editor you can use to enter and update your source and procedure members. This information is contained in *Application Development Tools: Source Entry Utility User's Guide and Reference, SC09-1338*.

Conventions in Format Notation

In COBOL, basic formats are prescribed for the various elements of the language. In this manual, these formats are presented in a uniform system of notation that is explained in the following paragraphs. This notation is designed to assist the programmer in writing COBOL source statements.

- Reserved words are printed entirely in CAPITAL LETTERS. These words have preassigned meanings in COBOL. If any reserved word is misspelled, it is not recognized as a reserved word and can cause an error in the program. The two types of reserved words are keywords and optional words.
  - Keywords are required by the syntax of the format unless the portion of the format containing them is optional. In formats, keywords are shown in UNDERLINED CAPITAL LETTERS. A missing keyword is considered an error in the program.
  - Optional words are included only for readability. They can be included or omitted without changing the syntax of the program. Optional words are CAPITALIZED but not underlined.
- Words printed in lowercase letters represent information to be supplied by the user and are defined in the *System/38-Compatible COBOL User's Guide and Reference*, *SC09-1814*.

About This Summary

- For easier reference, some user-defined words are followed be a hyphen and a digit or letter. This suffix does not change the syntactical definition of the word.
- Braces ({ }) enclosing listed items indicate (1) that exactly one of the enclosed stacked items must be specified, and/or (2) when followed by an ellipsis, that the enclosed unit or item must be specified at least once.
- Square brackets ([ ]) indicate that the enclosed item or unit can be used or omitted, as requested for the program. When two or more items are stacked within brackets, one or none of them can be specified. When followed by an ellipsis, the item or unit can be repeated.
- The ellipsis (...) indicates that the immediately preceding unit can occur once or any number of times in succession. A unit can be a single lowercase word or a group of lowercase words and one or more reserved words enclosed in brackets and/or braces. When repetition is used, everything enclosed within the immediately preceding brackets or braces must be repeated.
- The arithmetic and logical operators (+, –, <, >, =) that appear in formats are required although they are not underlined.
- All punctuation and other special characters appearing in formats (except braces, brackets, ellipsis, commas, and semicolons) are required by the syntax of the format when they are shown; if they are omitted, an error occurs in the program. Additional punctuation can be specified, according to the punctuation rules given later in this manual.
- The required clauses and (when written) optional clauses must be written in the sequence shown in the format except where the accompanying text states otherwise.
- Comments, restrictions, and clarifications on the use and meaning of every format are contained in the *System/38-Compatible COBOL User's Guide and Reference*.
-

IBM extensions to American National Standard, COBOL, X3.23-1974, that are part of a command syntax are boxed like this sentence.

About This Summary

-

```
****************************************************************
* COBOL clauses and statements that are syntax-checked, but are *
* treated as documentation by the System/38-Compatible COBOL    *
* compiler, are boxed like this sentence.                       *
****************************************************************
```

- IBM extensions

IBM Extension

IBM extensions to American National Standard (ANS) COBOL, X3.23-1974, that are part of the text description begin with the paragraph heading, **IBM Extension** and are separated from the regular text as is this paragraph.

End of IBM Extension

About This Summary

The System/38-Compatible COBOL syntax diagram below shows this standard format notation.

Format 2

```
SELECT file-name
                              ******************************
     ASSIGN TO assignment-name-1 * [ , assignment-name-2 ] ... *
                              ******************************

  ***********************************
  * [ RESERVE integer-1 [ AREA  ] ] *
  *                     [ AREAS ]   *
  ***********************************

     ORGANIZATION IS INDEXED

   [                  { SEQUENTIAL } ]
   [ ACCESS MODE IS { RANDOM     }   ]
   [                { DYNAMIC    }   ]

     RECORD KEY IS { EXTERNALLY-DESCRIBED-KEY } [ WITH DUPLICATES ]
                   { data-name-2              }

    [ FILE STATUS IS data-name-1 ] .
```

You must write the required clauses and the optional clauses (when written) in the order shown in the format.

Comments, restrictions, and clarifications on the use and meaning of every format are contained in the descriptions in the *System/38-Compatible COBOL User's Guide and Reference*.

Chapter 1. Compiling a Program

Create COBOL Program Command

To compile a System/38-Compatible COBOL source program into an object program, you must enter the CRTCBLPGM (Create COBOL Program) command. This invokes the System/38-Compatible COBOL compiler. The command is valid in batch and interactive jobs, or from other programs in the System/38 environment. You need QTEMP on the library list, but QCBL is no longer required.

Note: CRTCBLPGM is used to create System/38-Compatible COBOL programs in the System/38 environment.

If the COBOL compiler terminates, the escape message CBL9001 is issued. A CL program can monitor for this exception by using the CL command MONMSG (Monitor Message).

All object names specified and entered with the CRTCBLPGM command must be composed of alphanumeric characters, the first of which must be alphabetic. The length of the names cannot exceed 10 characters. See the *CL Reference* manual for a detailed description of object naming rules and for a complete description of the CL command syntax.

When the CRTCBLPGM command is issued in a CL program, concatenation expressions can be used for all parameter values. See the *CL Reference* manual for more information about concatenation expressions.

Note: The number of entries in the Object Definition Table (ODT) and the amount of storage required by a COBOL program varies with the number and kinds of COBOL statements used in the program. A combination of these factors can cause the AS/400 system internal size limits for the program to be exceeded. If this occurs, use the *NOUNREF option of the GENOPT parameter. If the problem persists, the program must be rewritten.

© Copyright IBM Corp. 1994

Compiling a Program

When the *NOUNREF option is specified, only names that are referenced or are needed for data structuring are defined. This option is useful when the program contains many unreferenced identifiers.

If member type is not CBL38, a warning will be issued.

If you key in the CRTCBLPGM command and press the F4 key, the following screen is shown on the display.

```
CRTCBLPGM                  Create COBOL Program

Type choices, press Enter.

Program  . . . . . .  *PGMID      Name, *PGMID
  Library  . . . . .    QGPL      Name
Source file  . . . .  QCBLSRC     Name
  Library  . . . . .    *LIBL     Name, *LIBL
Source member  . . .  *PGM        Name, *PGM
Source listing options .  .  .    *SOURCE, *NOSOURCE, *SRC...
            + for more values
Generation options . . . . .      *NOLIST, *LIST, *NOXREF...
            + for more values
Generation severity level  . .  29          0-29
Print file . . . . . .  QSYSPRT    Name
  Library  . . . . .    *LIBL     Name, *LIBL
FIPS flagging level  . . . .  *NO          *NO, *L, *LI, *HI, *H
Flagging severity  . . . . .  0           0-99
User profile . . . . . .  *USER        *USER, *OWNER
Public authority . . . . .  *NORMAL      *NORMAL, *ALL, *NONE
                                                              More...
F3=Exit   F4=List   F5=Refresh   F11=Keywords   F12=Previous   F13=Prompter help
```

Figure 1. The First CRTCBLPGM Prompt Screen

Each parameter on the screen displays a default value. Move the cursor past items where you want the default value to apply. Type over any items where you want to set a different value or option.

You must enter values for the library and program name by which the compiled program is to be known, and the name of the source file that contains the source program to be compiled.

Scroll Up to display the next screen of additional parameters.

```
CRTCBLPGM                  Create COBOL Program

Type choices, press Enter.

Text 'description' . . . . . .   *SRCMBRTXT
_____

Compiler debugging dump
                                 1_____          1-32767, *
                                 32767_          1-32767
Intermediate text dump . . . . .  0_____         0-31

                                                                    Bottom
F3=Exit   F4=List   F5=Refresh   F11=Keywords   F12=Previous   F13=Prompter help
```

Figure 2. The Second CRTCBLPGM Prompt Screen

Note: Any CRTCBLPGM command default can be changed by using the CL command CHGCMDDFT. For more information, refer to the *CL Reference*.

If these parameter values are acceptable, press ENTER to process the command.

Press F3 to exit without processing the command.

Press F11 from either screen to display the corresponding parameter keywords and entry fields, as shown below.

Compiling a Program

```
CRTCBLPGM                  Create COBOL Program

Type choices, press Enter.

Program  . . . . . . . . . . . . PGM           *PGMID
  Library  . . . . . . . . . . .                 QGPL
Source file  . . . . . . . . . . SRCFILE        QCBLSRC
  Library  . . . . . . . . . . .                 *LIBL
Source member  . . . . . . . . . SRCMBR         *PGM
Source listing options . . . . . OPTION
                          + for more values
Generation options . . . . . . . GENOPT
                          + for more values
Generation severity level  . . . GENLVL         29
Print file . . . . . . . . . . . PRTFILE        QSYSPRT
  Library  . . . . . . . . . . .                 *LIBL
FIPS flagging level  . . . . . . FIPS           *NO
Flagging severity  . . . . . . . FLAG           0
User profile . . . . . . . . . . USRPRF         *USER
Public authority . . . . . . . . PUBAUT         *NORMAL
                                                                More...
F3=Exit   F4=List   F5=Refresh   F11=Choices   F12=Previous   F13=Prompter help
```

Figure 3. The First CRTCBLPGM Prompt Screen, Showing Keywords

```
CRTCBLPGM                  Create COBOL Program

Type choices, press Enter.

Text 'description' . . . . . . . TEXT          *SRCMBRTXT
_____________________

Compiler debugging dump          DUMP
                                               1
                                               32767
Intermediate text dump . . . . . ITDUMP        0









                                                                Bottom
F3=Exit   F4=List   F5=Refresh   F11=Choices   F12=Previous   F13=Prompter help
```

Figure 4. The Second CRTCBLPGM Prompt Screen, Showing Keywords

The Run-Time Switch

The run-time switch dynamically activates the debugging code that is generated when `WITH DEBUGGING MODE` is specified.

Two commands are provided to control the run-time switch. To set the run-time switch on, enter the command `ENTCBLDBG`. To display the prompt screen, press the F4 key immediately after entering the command. The following screen will be displayed.

```
ENTCBLDBG                   Enter COBOL Debug

Type choices, press Enter.

Program  . . . . . .  _________   Name
                       *LIBL      Name, *LIBL

                                                                  Bottom
F3=Exit   F4=List   F5=Refresh   F11=Keywords   F12=Previous   F13=Prompter help
```

To view the keywords associated with the parameter and option shown above, press the F11 key. The following screen will be displayed.

Compiling a Program

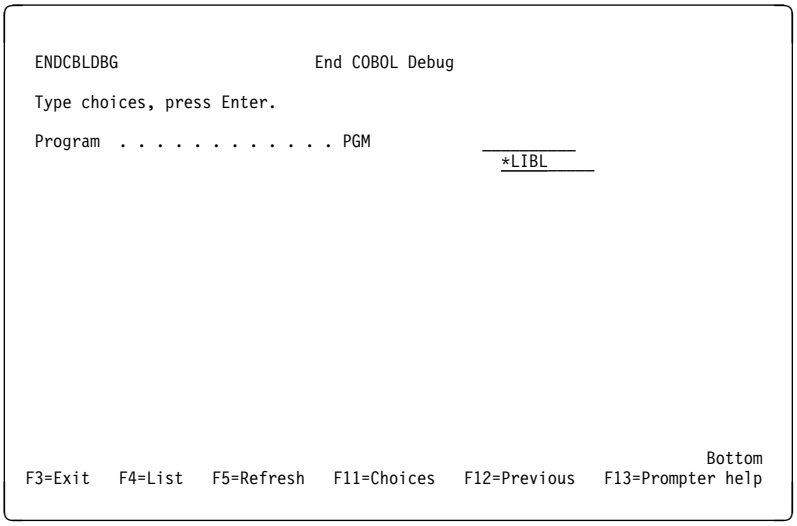
```
ENTCBLDBG                   Enter COBOL Debug

Type choices, press Enter.

Program  . . . . . . . . . . . . PGM          _________
                                               *LIBL

                                                                      Bottom
F3=Exit   F4=List   F5=Refresh   F11=Choices   F12=Previous   F13=Prompter help
```

To set the run-time switch off, enter the command ENDCBLDBG.

As indicated above, the prompt screen will be displayed if you press the F4 key.

```
ENDCBLDBG                    End COBOL Debug

Type choices, press Enter.

Program  . . . . . . . . . . . .  _________    Name
                                   *LIBL       Name, *LIBL

                                                                      Bottom
F3=Exit   F4=List   F5=Refresh   F11=Keywords   F12=Previous   F13=Prompter help
```

Similarly, By pressing the F11 key the following keyword screen will be displayed.



The default for the run-time switch is off.

PROCESS Statement

The PROCESS statement specifies compile-time options for the COBOL source program. This statement is optional. If you consistently want any compiler options that are not default options included as part of the source program, you must precede the Identification Division header with this statement. Options specified in the PROCESS statement override the corresponding options specified in the CRTCBLPGM command.

Format

```
PROCESS option-1 [ option-2 ] . . . [ option-n ] [ . ]
```

Note: A COPY statement can be specified within a PROCESS statement.

Compiling a Program

Chapter 2. Identification Division

The IDENTIFICATION DIVISION Statement

Format

```
  IDENTIFICATION DIVISION.
  PROGRAM-ID. program-name.
[ AUTHOR. [ comment-entry ] . . . ]
[ INSTALLATION. [ comment-entry ] . . . ]
[ DATE-WRITTEN. [ comment-entry ] . . . ]
[ DATE-COMPILED. [ comment-entry ] . . . ]
[ SECURITY. [ comment-entry ] . . . ]
```

© Copyright IBM Corp. 1994

Identification Division

Chapter 3. Environment Division

The ENVIRONMENT DIVISION Statement

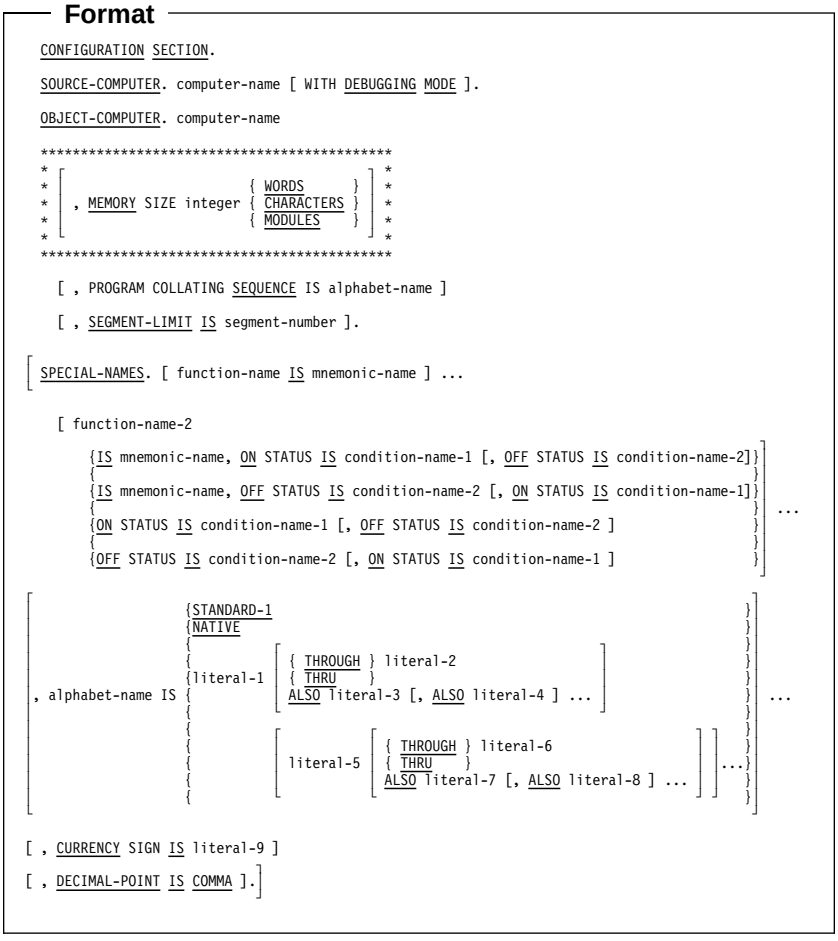
Format

```
  ENVIRONMENT DIVISION.

  CONFIGURATION SECTION.

  SOURCE-COMPUTER. source-computer-entry

  OBJECT-COMPUTER. object-computer-entry

[ SPECIAL-NAMES. special-names-entry ]

[ INPUT-OUTPUT SECTION.

  FILE-CONTROL. { file-control-entry } . . .

[ I-O-CONTROL. input-output-control-entry ] ]
```

© Copyright IBM Corp. 1994

Configuration Section

Format



Input-Output Section

The keyword FILE-CONTROL appears only once–at the beginning of the paragraph and before the first file-control entry. The keyword I-O-CONTROL appears only once–at the beginning of the paragraph and before the input-output-control entry.

Format

```
[ INPUT-OUTPUT SECTION.
  FILE-CONTROL. { file-control-entry } . . .

[ I-O-CONTROL. input-output-control-entry ] ] .
```

FILE-CONTROL Paragraph–Sequential File Entries (READER, PUNCH, PUNCHPRINT, PRINT, PRINTER, TAPEFILE, DISKETTE, FORMATFILE, DISK, DATABASE)

Format 1

```
SELECT [ OPTIONAL ] file-name
                                ******************************
      ASSIGN TO assignment-name-1 * [ , assignment-name-2 ] ... *
                                ******************************

  ***********************************
  * [ RESERVE integer-1 [ AREA  ] ] *
  *                     [ AREAS ] ] *
  ***********************************

     [ ORGANIZATION IS SEQUENTIAL ]

     [ ACCESS MODE IS SEQUENTIAL ]

     [ FILE STATUS IS data-name-1 ] .
```

FILE-CONTROL Paragraph–Indexed File Entries (DISK, DATABASE)

Format 2

```
SELECT file-name
                                ******************************
     ASSIGN TO assignment-name-1 * [ , assignment-name-2 ] ... *
                                ******************************

  **********************************
  * [                    [      ] ] *
  * | RESERVE integer-1  | AREA | | *
  * |                    | AREAS | | *
  * [                    [      ] ] *
  **********************************

     ORGANIZATION IS INDEXED

   [                  { SEQUENTIAL } ]
   | ACCESS MODE IS { RANDOM     } |
   [                  { DYNAMIC    } ]

     RECORD KEY IS { EXTERNALLY-DESCRIBED-KEY } [ WITH DUPLICATES ]
                   { data-name-2              }

   [ FILE STATUS IS data-name-1 ] .
```

FILE-CONTROL Paragraph–Relative (Direct) File Entries (DISK, DATABASE)

Format 3

```
SELECT file-name
                                ******************************
     ASSIGN TO assignment-name-1 * [ , assignment-name-2 ] ... *
                                ******************************

  ***********************************
  * [ RESERVE integer-1 [ AREA  ] ] *
  *                     [ AREAS ]   *
  *                                 *
  ***********************************

     ORGANIZATION IS RELATIVE

   [                 { SEQUENTIAL [ , RELATIVE KEY IS data-name-3 ] }
   [                 {                                              }
   [ ACCESS MODE IS { { RANDOM  } , RELATIVE KEY IS data-name-3    }
   [                 { { DYNAMIC }                                  }


   [ FILE STATUS IS data-name-1 ] .
```

FILE-CONTROL Paragraph–Sort or Merge File Entries

Format 4

```
                ********************************************************
SELECT file-name * ASSIGN TO assignment-name-1 [ , assignment-name-2 ] ... * .
                ********************************************************
```

FILE-CONTROL Paragraph–TRANSACTION File Entries (WORKSTATION)

Format 5

```
SELECT file-name
                              ********************************
     ASSIGN TO assignment-name-1 * [ , assignment-name-2 ] ... *
                              ********************************

     ORGANIZATION IS TRANSACTION

   [ ACCESS MODE IS { SEQUENTIAL                          } ]
                    { DYNAMIC, RELATIVE KEY IS data-name-3 }

   [ FILE STATUS IS data-name-1 { , data-name-5 } ]

   [ CONTROL-AREA is data-name-6 ] .
```

I-O-CONTROL Paragraph

Format

```
I-O-CONTROL.

**********************************************************
* [ RERUN ON assignment-name                             *
*                                                        *
*   EVERY integer-1 RECORDS OF file-name-1 ] . . .       *
**********************************************************

[ SAME [ RECORD ]                                        ]
       [ SORT   ] AREA FOR file-name-2 { , file-name-3 } . . . . . .
       [ SORT-MERGE ]                                    ]

************************************************************
*                                                          *
* [ MULTIPLE FILE TAPE CONTAINS                            *
*                                                          *
*   file-name-4 [ POSITION integer-2 ]                     *
*                                                          *
*                                                          *
*   [ file-name-5 [ POSITION integer-3 ] ] . . . ] . . .   *
*                                                          *
************************************************************

[ COMMITMENT CONTROL FOR file-name-6                       ]
                                                           ]
[ , file-name-7 ]  . . .                                   ]
                                                       .   ]
```

Environment Division

Chapter 4. Data Division

File Section

The order of the clauses specified after the `FD file-name` or `SD file-name` clause is optional.

FD Entry–Files (FORMATFILE, DATABASE, DISK, READER, PUNCH, PUNCHPRINT, PRINT)

Format 1

```
[ FD file-name

    [ BLOCK CONTAINS [ integer-1 TO ] integer-2 { RECORDS    } ]
                                                { CHARACTERS } ]

    [ RECORD CONTAINS [ integer-3 TO ] integer-4 CHARACTERS ]

  *******************************************************
  *    LABEL { RECORD IS   } { STANDARD }               *
  *          { RECORDS ARE } { OMITTED  }               *
  *     [                                               *
  *     | VALUE OF user-name-1 IS { data-name-1 }       *
  *     |                        { literal-1   }       *
  *     [                                               *
  *           [                                   ]     *
  *           | , user-name-2 IS { data-name-2 } | ... | *
  *           |                 { literal-2   } |     | *
  *           [                                   ]     *
  *******************************************************

    [ DATA { RECORD IS   } data-name-3 [ , data-name-4 ] ... ] .
           { RECORDS ARE }                                    ]

{ record-description-entry } ... ] ...
```

© Copyright IBM Corp. 1994

FD Entry–Files (DISKETTE)

Format 2

```
[ FD file-name

    [ BLOCK CONTAINS [ integer-1 TO ] integer-2 { RECORDS    } ]
                                                { CHARACTERS }

    [ RECORD CONTAINS [ integer-3 TO ] integer-4 CHARACTERS ]

  ********************************************************
  *    LABEL { RECORD IS   } { STANDARD }                *
  *          { RECORDS ARE } { OMITTED  }                *
  *    [                                                 *
  *    [ VALUE OF user-name-1 IS { data-name-1 }         *
  *    [                         { literal-1   }         *
  *    [                                                 *
  *          [                                       ]   *
  *          [ , user-name-2 IS { data-name-2 } ] ... ]  *
  *          [                  { literal-2   } ]     ]  *
  ********************************************************

    [ DATA { RECORD IS   } data-name-3 [ , data-name-4 ] ... ]
           { RECORDS ARE }

    [ CODE-SET is alphabet-name ] .

{ record-description-entry } ... ] ...
```

FD Entry–Files (TAPEFILE)

Format 3

```
[ FD file-name

    [ BLOCK CONTAINS [ integer-1 TO ] integer-2 { RECORDS    } ]
                                                { CHARACTERS } ]

    [ RECORD CONTAINS [ integer-3 TO ] integer-4 CHARACTERS ]

      LABEL { RECORD IS   } { STANDARD }
            { RECORDS ARE } { OMITTED  }
  ******************************************************
  *     [ VALUE OF user-name-1 IS { data-name-1 }      ]*
  *     [                         { literal-1   }      ]*
  *                                                    *
  *           [ , user-name-2 IS { data-name-2 } ] ... ]*
  *           [                  { literal-2   } ]     ]*
  ******************************************************

    [ DATA { RECORD IS   } data-name-3 [ , data-name-4 ] ... ]
           { RECORDS ARE }                                   ]

    [ CODE-SET is alphabet-name ] .

{ record-description-entry } ... ] ...
```

FD Entry–Files (PRINTER)

Format 4

```
[ FD file-name

    [ BLOCK CONTAINS [ integer-1 TO ] integer-2 { RECORDS    } ]
                                                { CHARACTERS } ]

    [ RECORD CONTAINS [ integer-3 TO ] integer-4 CHARACTERS ]

  ******************************************************
  *   LABEL { RECORD IS   } { STANDARD }               *
  *         { RECORDS ARE } { OMITTED  }               *
  *                                                    *
  *                                                    *
  *     [ VALUE OF user-name-1 IS { data-name-1 }      *
  *     [                         { literal-1   }      *
  *                                                    *
  *           [ , user-name-2 IS { data-name-2 } ] ... ] *
  *           [                  { literal-2   } ]     ] *
  ******************************************************

    [ DATA { RECORD IS   } data-name-3 [ , data-name-4 ] ... ]
    [      { RECORDS ARE }                                   ]

    [ LINAGE IS { data-name-5 } LINES [ , WITH FOOTING AT { data-name-6 } ] ]
    [           { integer-5   }       [                   { integer-6   } ] ]

    [ , LINES AT TOP { data-name-7 } ] [ , LINES AT BOTTOM { data-name-8 } ] ] .
    [                { integer-7   } ] [                   { integer-8   } ] ]

{ record-description-entry } ... ] ...
```

FD Entry–TRANSACTION File

Format 5

```
[ FD file-name

  [ RECORD CONTAINS [ integer-3 TO ] integer-4 CHARACTERS ]

  ************************************
  * LABEL { RECORD IS   } { OMITTED  } *
  *       { RECORDS ARE } { STANDARD } *
  *                                    *
  ************************************

  [ DATA { RECORD IS   } data-name-3 [ , data-name-4 ] . . . ] .
         { RECORDS ARE }

{ record-description-entry } . . . ]
```

SD Entry

Format 6

```
[ SD file-name

     [ RECORD CONTAINS [ integer-1 TO ] integer-2 CHARACTERS ]

     [ DATA { RECORD IS   } data-name-1 [ , data-name-2 ] ... ] .
            { RECORDS ARE }

{ record-description-entry } ... ]
```

Working-Storage Section

The clauses of the data description entry can be specified in any order except that the `level-number` and data-name/`FILLER` clause must be specified first, and if the `REDEFINES` clause is specified, it must immediately follow the data-name/`FILLER` clause.

Data Description Entry–Format 1–Item Description

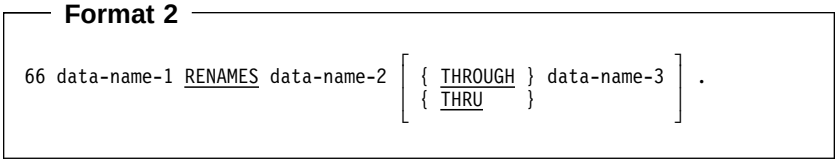
Format 1

```
level-number { data-name-1 }
             { FILLER      }

      [ REDEFINES data-name-2 ]

      [ { PICTURE } IS character-string ]
      [ { PIC    }                      ]

      [                                 ]
      [              {   DISPLAY          } ]
      [              {   COMPUTATIONAL    } ]
      [              {   COMP             } ]
      [              {                    } ]
      [ [ USAGE IS ] {  COMPUTATIONAL-3   } ]
      [              {  COMP-3            } ]
      [              {  COMPUTATIONAL-4   } ]
      [              {  COMP-4            } ]
      [              {                    } ]
      [              {   INDEX            } ]
      [                                 ]

      [ [ SIGN IS ] { LEADING  } [ SEPARATE CHARACTER ] ]
      [             { TRAILING }                        ]

      [ OCCURS { integer-1 TO integer-2 TIMES DEPENDING ON data-name-3 } ]
      [        { integer-2 TIMES                                       } ]

               [ { ASCENDING  } KEY IS data-name-4 [ , data-name-5 ] . . . ] . . .
               [ { DESCENDING }                                            ]

               [                                                           ]
               [ INDEXED BY index-name-1 [ , index-name-2 ] . . . ]
               [                                                           ]

     **********************************
     * [ { SYNCHRONIZED } [ LEFT  ] ] *
     * [ { SYNC         } [ RIGHT ] ] *
     **********************************

      [ { JUSTIFIED } RIGHT ]
      [ { JUST      }       ]

      [ BLANK WHEN ZERO ]

      [ VALUE IS literal ] .
```

The clauses of the data description entry can be specified in any order except that the level-number and data-name/FILLER clause must be specified first, and if the REDEFINES clause is specified, it must immediately follow the data-name/FILLER clause.

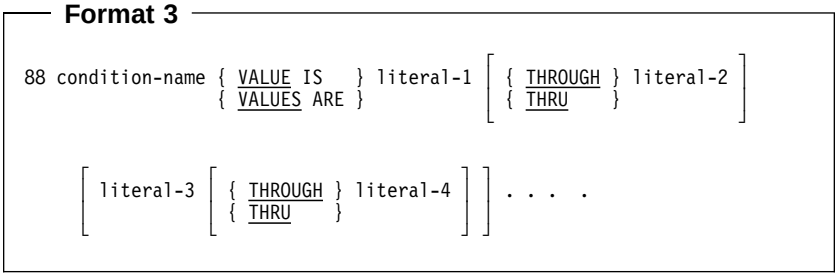
Data Description Entry–Format 2–Regroup or Rename Data Items

Format 2



Data Description Entry–Format 3–Condition-Name Description

Format 3



Data Description Entry–Format 4–Boolean Data Description

Format 4

```
level-number { data-name-1 }
             { FILLER      }

      [ REDEFINES data-name-2 ]

      [ { PICTURE } IS 1 ]
      [ { PIC    }       ]

      [ [ USAGE IS ] DISPLAY ]

      [ OCCURS { integer-1 TO integer-2 TIMES DEPENDING ON data-name-3 }
               { integer-2 TIMES                                       }

               [ INDEXED BY index-name-1 [ , index-name-2 ] . . . ] ]

      [ { INDICATOR  }              ]
      [ { INDICATORS } integer-3    ]
      [ { INDIC      }              ]

     **********************************
     * [                            ] *
     * [ { SYNCHRONIZED } [ LEFT  ] ] *
     * [ { SYNC         } [ RIGHT ] ] *
     * [                            ] *
     *                                *
     * [ { JUSTIFIED } RIGHT ]        *
     * [ { JUST      }       ]        *
     *                                *
     **********************************

      [ VALUE IS Boolean-literal ] .
```

Linkage Section

Any data description entry clause, except the VALUE clause, can describe items in the Linkage Section. The VALUE clause can be specified only for level-88 items. See “Working-Storage Section” on page 24 for data description entry clause formats.

Chapter 5. Procedure Division

The PROCEDURE DIVISION Statement

Procedure Division–Format 1–Section, Declaratives

Format 1

```
  PROCEDURE DIVISION [ USING data-name-1 [ , data-name-2 ] . . . ] .

[ DECLARATIVES.

{ section-name SECTION [ segment-number ] . use-sentence.
[ paragraph-name. [ sentence ] ... ] . . . } . . .

  END DECLARATIVES. ]

{ section-name SECTION [ segment-number ] .
[ paragraph-name. [ sentence ] ... ] . . . } . . .
```

Procedure Division–Format 2

Format 2

```
  PROCEDURE DIVISION [ USING data-name-1 [ , data-name-2 ] . . . ] .

{ paragraph-name. [ sentence ] . . . } . . .
```

© Copyright IBM Corp. 1994

Procedure Division Statements

ACCEPT Statement–Format 1

Format 1

```
ACCEPT identifier [ FROM mnemonic-name ]
```

ACCEPT Statement–Format 2

Format 2

```
                       { DATE }
ACCEPT identifier FROM { DAY  }
                       { TIME }
```

ACCEPT Statement–Format 3

Format 3

```
ACCEPT identifier FROM mnemonic-name
  [ FOR file-name ]
```

ACCEPT Statement–Format 4–Local Data Area

Format 4

```
ACCEPT identifier-1 FROM mnemonic-name

****************************
*  [                       ] *
*  [ FOR { identifier-2 } ] *
*  [     { literal      } ] *
*  [                       ] *
****************************
```

ACCEPT Statement–Format 5–TRANSACTION Attributes

Format 5

```
ACCEPT identifier-1  FROM mnemonic-name

[                                                ]
[ FOR { identifier-2 } [ FOR file-name ] ]  .
[     { literal      }                   ]
[                                                ]
```

ACQUIRE Statement–TRANSACTION File

Format

```
ACQUIRE { identifier } FOR file-name
        { literal    }
```

ADD Statement–Format 1

Format 1

```
ADD { identifier-1 } [ , identifier-2 ] . . . TO identifier-m [ ROUNDED ]
    { literal-1    }   , literal-2

    [ , identifier-n [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]
```

ADD Statement–Format 2–Giving

Format 2

```
ADD { identifier-1 } { identifier-2 } [ , identifier-3 ] . . .
    { literal-1    } { literal-2    }   , literal-3

    GIVING identifier-m [ ROUNDED ] [ , identifier-n [ ROUNDED ] ] . . .

    [ ON SIZE ERROR imperative-statement ]
```

ADD Statement–Format 3–Corresponding

Format 3

```
ADD { CORRESPONDING } identifier-1 TO identifier-2 [ ROUNDED ]
    { CORR          }

    [ ON SIZE ERROR imperative-statement ]
```

ALTER Statement

Format

```
ALTER procedure-name-1 TO [ PROCEED TO ] procedure-name-2

    [ , procedure-name-3 TO [ PROCEED TO ] procedure-name-4 ]  . . .
```

CALL Statement

Format

```
CALL { identifier-1 } [ USING data-name-1 [ . data-name-2 ] . . . ]
     { literal-1    }

     [ ON OVERFLOW imperative-statement ]
```

CANCEL Statement

Format

```
CANCEL { identifier-1 } [ , identifier-2 ] . . .
       { literal-1    } [ , literal-2    ]
```

Procedure Division

CLOSE Statement–Format 1

Format 1

```
                    [ { REEL } [ WITH NO REWIND ] ]
                    [ { UNIT } [ FOR REMOVAL    ] ]
CLOSE file-name-1   [                             ]
                    [   WITH { NO REWIND }        ]
                    [        { LOCK      }        ]

  [                 [ { REEL } [ WITH NO REWIND ] ] ]
  [                 [ { UNIT } [ FOR REMOVAL    ] ] ]
  [ , file-name-2   [                             ] ]  . . .
  [                 [   WITH { NO REWIND }        ] ]
  [                 [        { LOCK      }        ] ]
```

CLOSE Statement–Format 2–TRANSACTION File

Format

```
CLOSE file-name-1 [ WITH LOCK ]

  [ file-name-2 [ WITH LOCK ] ] . . .
```

COMMIT Statement

Format

```
COMMIT
```

COMPUTE Statement

Format

```
COMPUTE identifier-1 [ ROUNDED ] [ , identifier-2 [ ROUNDED ] ] . . .

        = arithmetic-expression [ ON SIZE ERROR imperative-statement ]
```

DELETE Statement

Format

```
DELETE file-name RECORD [ [ FORMAT IS { identifer } ] ]
                        [             { literal   } ]

     [ INVALID KEY imperative-statement ]
```

DISPLAY Statement–Format 1

Format 1

```
DISPLAY { identifier-1 } [ , identifier-2 ]
        { literal-1    } [ , literal-2    ]

        ... [ UPON mnemonic-name ]
```

DISPLAY Statement–Format 2–Local Data Area

Format 2

```
DISPLAY { identifier-1 } [ { , identifier-2 } ] . . .
        { literal-1    }   { , literal-2    }

  UPON   mnemonic-name

**************************
* [ FOR { identifier-3} ] *
* [     { literal-3   } ] *
**************************
```

DIVIDE Statement–Format 1

Format 1

```
DIVIDE { identifier-1 } INTO identifier-2 [ ROUNDED ]
       { literal-1    }

    [ , identifier-3 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]
```

DIVIDE Statement–Format 2–Giving

Format 2

```
DIVIDE { identifier-1 } { INTO } { identifier-2 } GIVING identifier-3 [ ROUNDED ]
       { literal-1    } { BY   } { literal-2    }

    [ , identifier-3 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]
```

DIVIDE Statement–Format 3–Giving, Remainder

Format 3

```
DIVIDE { identifier-1 } { INTO } { identifier-2 } GIVING identifier-3 [ ROUNDED ]
       { literal-1    } { BY   } { literal-2    }

    REMAINDER identifer-4 [ ON SIZE ERROR imperative-statement ]
```

DROP Statement–TRANSACTION File

Format

```
DROP { identifier } FROM file-name
     { literal    }
```

ENTER Statement

Format

```
*****************************************
* ENTER language-name [ routine-name ] . *
*****************************************
```

EXIT Statement

Format

```
EXIT [ PROGRAM ].
```

GO TO Statement–Format 1

Format 1

```
GO TO [ procedure-name-1 ]
```

GO TO Statement–Format 2–Depending On (Conditional)

Format 2

```
GO TO procedure-name-1 [ , procedure-name-2 ] . . . , procedure-name-n

   DEPENDING ON identifier
```

IF Statement

Format

```
IF condition [ THEN ] { statement-1    } [ { ELSE statement-2    } ]
                      { NEXT SENTENCE  } [ { ELSE NEXT SENTENCE  } ]
```

INSPECT Statement–Format 1–Tallying

Format 1

```
INSPECT identifier-1 TALLYING

    {                    {  { { ALL     } { identifier-3 } }
    { , identifier-2 FOR { ,{ { LEADING } { literal-1    } }
    {                    {  { CHARACTERS                  }
    {                    {

        [                                          ] }       }
        [ { BEFORE } INITIAL { identifier-4 } ] } . . . } . . .
        [ { AFTER  }         { literal-2    } ] }       }
        [                                          ] }       }
```

INSPECT Statement–Format 2–Replacing

Format 2

```
INSPECT identifier-1 REPLACING

    {
    { CHARACTERS BY { identifier-6 } [ { BEFORE } INITIAL { identifier-7 } ]
    {               { literal-4    } [ { AFTER  }         { literal-5    } ]
    {
    { {
    { {   { ALL     } {   { identifier-5 } BY { identifier-6 }
    { { , { LEADING } { , { literal-3    }    { literal-4    }
    { {   { FIRST   } {
    { {             {

                                                        }
                                                  }     }
                                                  }     }
        [                                          ] }     }     }
        [ { BEFORE } INITIAL { identifier-7 } ] } ... } ... }
        [ { AFTER  }         { literal-5    } ] }     }     }
        [                                          ] }     }     }
                                                        }
                                                        }
```

INSPECT Statement–Format 3–Tallying and Replacing

Format 3

```
INSPECT identifier-1 TALLYING

    {
    {                    {
    { , identifier-2 FOR {   { { ALL     } { identifier-3 } }
    {                    { , { { LEADING } { literal-1    } }
    {                    {   { CHARACTERS                  }
    {
    {

                                                          }
                                                          }
       [ { BEFORE } INITIAL { identifier-4 } ] }      }
       [ { AFTER  }         { literal-2    } ] } . . . } . . .
                                                 }      }
                                                 }      }
                                                 }

    REPLACING

     {                                                                  }
     { CHARACTERS BY { identifier-6 } [ { BEFORE }   INITIAL { identifier-7 } ] }
     {               { literal-4    } [ { AFTER  }           { literal-5    } ] }
     {                                                                  }
     { {                {                                               }
     { {   { ALL     } {   { identifier-5 } BY { identifier-6 }         }
     { { , { LEADING } { , { literal-3    }    { literal-4    }         }
     { {   { FIRST   } {                                                }
     {                                                                  }
     {    [                                    ] }     }                }
     {    [ { BEFORE } INITIAL { identifier-7 } ] } ... } ...           }
     {    [ { AFTER  }         { literal-5    } ] }     }                }
     {                                             }     }            }
     {                                                                  }
     {                                                                  }
```

MERGE Statement

Format

```
MERGE file-name-1 ON { ASCENDING  } KEY data-name-1 [ , data-name-2 ] . . .
                     { DESCENDING }

              [ ON { ASCENDING  } KEY data-name-3 [ , data-name-4 ] . . . ] . . .
                 { DESCENDING }

    [ COLLATING SEQUENCE IS alphabet-name ]

      USING file-name-2, file-name-3 [ , file-name-4 ] . . .

    {                                                                    }
    { OUTPUT PROCEDURE IS section-name-1 [ { THROUGH } section-name-2 ] }
    {                                    [ { THRU    }                ] }
    {                                                                    }
    { GIVING file-name-5                                                 }
```

MOVE Statement–Format 1

Format 1

```
MOVE { identifier-1 } TO identifier-2 [ , identifier-3 ] . . .
     { literal      }
```

MOVE Statement–Format 2–Corresponding

Format 2

```
MOVE { CORRESPONDING } identifier-1 TO identifier-2
     { CORR          }
```

MULTIPLY Statement–Format 1

Format 1

```
MULTIPLY { identifier-1 } BY identifier-2 [ ROUNDED ]
         { literal-1    }
    [ , identifier-3 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]
```

MULTIPLY Statement–Format 2–Giving

Format 2

```
MULTIPLY { identifier-1 } BY { identifier-2 } GIVING identifier-3 [ ROUNDED ]
         { literal-1    }    { literal-2    }
    [ , identifier-4 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]
```

OPEN Statement–Sequential Files

Format 1

```
     {                                                    }
     { INPUT file-name-1 [ REVERSED          ]            }
     {                   [ WITH NO REWIND    ]            }
     {                                                    }
     {      [ , file-name-2 [ REVERSED          ] ] . . . }
     {      [               [ WITH NO REWIND    ] ]       }
OPEN {                                                    } . . .
     {                                                    }
     { OUTPUT file-name-3 [ WITH NO REWIND ]              }
     {                                                    }
     {      [ , file-name-4  [ WITH NO REWIND ] ] . . .   }
     {                                                    }
     { I-O file-name-5 [ , file-name-6 ] . . .            }
     {                                                    }
     { EXTEND file-name-7 [ , file-name-8 ] . . .         }
```

OPEN Statement–Indexed and Relative Files

Format 2

```
     {                                                  }
     { { INPUT  }                                       }
OPEN { { OUTPUT } file-name-1 [ , file-name-2 ] . . . } . . .
     { { I-O    }                                       }
     {                                                  }
```

OPEN Statement–TRANSACTION File

Format 1

```
OPEN I-O file-name-1 [ file-name-2 ] . . .
```

PERFORM Statement–Format 1

Format 1

```
PERFORM procedure-name-1  [ { THROUGH } procedure-name-2 ]
                          [ { THRU    }                  ]
```

PERFORM Statement–Format 2–Multiple Times

Format 2

```
PERFORM procedure-name-1  [ { THROUGH } procedure-name-2 ]  { identifier-1 } TIMES
                          [ { THRU    }                  ]  { integer-1    }
```

PERFORM Statement–Format 3–Until Condition Satisfied

Format 3

```
PERFORM procedure-name-1  [ { THROUGH } procedure-name-2 ]  UNTIL condition-1
                          [ { THRU    }                  ]
```

PERFORM Statement–Format 4–Varying Index or Identifier

Format 4

```
PERFORM procedure-name-1  [ { THROUGH } procedure-name-2 ]
                          [ { THRU    }                  ]

   VARYING { identifier-1 } FROM { identifier-2 }
           { index-name-1 }      { index-name-2 }
                                 { literal-2    }

   BY { identifier-3 } UNTIL condition-1
      { literal-3    }

   [
   [ AFTER { identifier-4 } FROM { identifier-5 }
           { index-name-4 }      { index-name-5 }
                                 { literal-5    }
   [

   BY { identifier-6 } UNTIL condition-2
      { literal-6    }

   [
   [ AFTER { identifier-7 } FROM { identifier-8 }
           { index-name-7 }      { index-name-8 }
                                 { literal-8    }
   [


                                              ] ]
   BY { identifier-9 } UNTIL condition-3      ] ]
      { literal-9    }                        ] ]
                                              ] ]
```

READ Statement–Format 1–Sequential Retrieval Using SEQUENTIAL Access

Format 1

```
READ file-name RECORD

  [ INTO identifier-1 ]

  [ FORMAT IS { identifier-2 } ]
              { literal-1    }

  [ AT END imperative-statement ]
```

READ Statement–Format 2–Sequential Retrieval Using DYNAMIC Access

Format 2

```
                { FIRST }
                { LAST  }
READ file-name  {       } RECORD
                { NEXT  }
                { PRIOR }

  [ INTO identifier-1 ]

  [ FORMAT IS { identifier-2 } ]
              { literal-1    }

  [ AT END imperative-statement ]
```

READ Statement–Format 3–Random Retrieval

Format 3

```
READ file-name RECORD [ INTO identifier-1 ]

     [ FORMAT IS { identifier-2 } ]
                 { literal-1    }

     [ INVALID KEY imperative-statement ]
```

READ Statement–Format 4–TRANSACTION File (Nonsubfile)

Format

```
READ file-name RECORD

    [ INTO identifier-1 ]

    [ FORMAT IS { identifier-2 } ]
                { literal-1    }

    [ TERMINAL IS { identifier-3 } ]
                  { literal-2    }

    [ { INDICATOR  [ IS  ] }                  ]
    [ { INDICATORS [ ARE ] } identifier-4     ]
    [ { INDIC              }                  ]

    [ NO DATA imperative-statement-1 ]

    [ AT END imperative-statement-2 ]
```

READ Statement–Format 5–TRANSACTION File (Subfile)

Format

```
READ SUBFILE file-name

    [ NEXT MODIFIED ] RECORD

    [ INTO identifier-1 ]

    [ FORMAT IS { identifier-2 } ]
                { literal-1    }

    [ TERMINAL IS { identifier-3 } ]
                  { literal-2    }

    [ { INDICATOR  [ IS  ] }                    ]
    [ { INDICATORS [ ARE ] } identifier-4       ]
    [ { INDIC              }                    ]


    [ INVALID KEY imperative-statement-1 ]

    [ AT END imperative-statement-2 ]
```

RELEASE Statement

Format

```
RELEASE record-name [ FROM identifier ]
```

RETURN Statement

Format

```
RETURN file-name RECORD [ INTO identifier ] AT END imperative-statement
```

REWRITE Statement–Format 1

Format 1

```
REWRITE record-name [ FROM identifier-1 ]

      [ FORMAT IS { identifier-2 } ]
                  { literal-1    }

     [ INVALID KEY imperative-statement ]
```

REWRITE Statement–Format 2–TRANSACTION File (Subfile)

Format 2

```
REWRITE SUBFILE record-name [ FROM identifier-1 ]

     FORMAT IS { identifier-2 }
               { literal-1    }

     [ TERMINAL IS { identifier-3 } ]
                   { literal-2    }

     [ { INDICATOR  [ IS  ] }                  ]
     [ { INDICATORS [ ARE ] } identifier-4     ]
     [ { INDIC              }                  ]

     [ INVALID KEY imperative-statement ]
```

ROLLBACK Statement

Format

```
ROLLBACK
```

SEARCH Statement–Format 1–Selective Table Search

Format 1

```
SEARCH identifier-1 [ VARYING { identifier-2 } ] [ AT END imperative-statement-1 ]
                    [         { index-name-1 } ]

    WHEN condition-1 { imperative-statement-2 }
                    { NEXT SENTENCE          }

  [ WHEN condition-2 { imperative-statement-3 } ] . . .
  [                  { NEXT SENTENCE          } ]
```

SEARCH Statement–Format 2–Key Table Search

Format 2

```
SEARCH ALL indentifier-1 [ AT END imperative-statement-1 ]

       { data-name-1 { IS EQUAL TO } { identifier-3            } }
       {             { IS =       } { literal-1              } }
  WHEN {                             { arithmetic-expression-1 } }
       { condition-name-1                                       }

       [     { data-name-2 { IS EQUAL TO } { identifier-4            } } ]
       [     {             { IS =       } { literal-2              } } ]
       [ AND {                            { arithmetic-expression-2 } } ] . . .
       [     {                                                        } ]
       [     { condition-name-2                                       } ]

       { imperative-statement-2 }
       { NEXT SENTENCE          }
```

SET Statement–Format 1–Switches

Format 1

```
SET mnemonic-name-1 [ , mnemonic-name-2 ] . . . TO { ON  }
                                                   { OFF }
```

SET Statement–Format 2–Condition Values

Format 2

```
SET condition-name-1 [ , condition-name-2 ] . . . TO TRUE
```

SET Statement–Format 3–Index Save/Restore

Format 3

```
SET { identifier-1 [ , identifier-2 ] . . . } TO { identifier-3 }
    { index-name-1 [ , index-name-2 ] . . . }    { index-name-3 }
                                                 { integer-1    }
```

SET Statement–Format 4–Index Adjustment

Format 4

```
SET index-name-4 [ , index-name-5 ] . . . { UP BY   } { identifier-4 }
                                          { DOWN BY } { integer-2    }
```

SORT Statement

Format

```
SORT file-name-1 ON { ASCENDING  } KEY data-name-1 [ , data-name-2 ] . . .
                    { DESCENDING }

              [ ON { ASCENDING  } KEY data-name-3 [ , data-name-4 ] . . . ] . . .
                 { DESCENDING }

   [ COLLATING SEQUENCE IS alphabet-name ]

   {                                                                      }
   { INPUT PROCEDURE IS section-name-1 [ { THROUGH } section-name-2 ]     }
   {                                   [ { THRU    }                ]     }
   {                                                                      }
   { USING file-name-2 [ , file-name-3 ] . . .                            }

   {                                                                      }
   { OUTPUT PROCEDURE IS section-name-3 [ { THROUGH } section-name-4 ]    }
   {                                    [ { THRU    }                ]    }
   {                                                                      }
   { GIVING file-name-4                                                   }
```

START Statement

Format

```
                  [        { EQUAL TO      }                                  ]
                  [        { =             }                                  ]
START file-name   [ KEY IS { GREATER THAN  } { EXTERNALLY-DESCRIBED-KEY     } ]
                  [        { >             } {                             } ]
                  [        { NOT LESS THAN } {   data-name-1 [ , data-name-2 ] . . . } ]
                  [        { NOT <         } {                             } ]

     [ FORMAT IS { identifier-1 } ]
                 { literal-1    }

   [ INVALID KEY imperative-statement ]
```

STOP Statement

Format

```
STOP { RUN     }
     { literal }
```

STRING Statement

Format

```
STRING { identifier-1 } [ , identifier-2 ] . . . DELIMITED BY { identifier-3 }
       { literal-1    } [ , literal-2    ]                    { literal-3    }
                                                              { SIZE         }

    [ , { identifier-4 } [ , identifier-5 ] . . . DELIMITED BY { identifier-6 } ]
      { literal-4    } [ , literal-5    ]                    { literal-6    }    . . .
    [                                                         { SIZE         } ]

    INTO identifier-7 [ WITH POINTER identifier-8 ]

    [ ON OVERFLOW imperative-statement ]
```

SUBTRACT Statement–Format 1

Format 1

```
SUBTRACT { identifier-1 } [ , identifier-2 ] . . . FROM identifier-3 [ ROUNDED ]
         { literal-1    } [ , literal-2    ]

   [ , identifier-4 [ ROUNDED ] ] . . . [ ON SIZE ERROR imperative-statement ]
```

SUBTRACT Statement–Format 2–Giving

Format 2

```
SUBTRACT { identifier-1 } [ , identifier-2 ] . . . FROM { identifier-3 }
         { literal-1    } [ , literal-2    ]          { literal-3    }

    GIVING identifier-4 [ ROUNDED ] [ , identifier-5 [ ROUNDED ] ] . . .

    [ ON SIZE ERROR imperative-statement ]
```

SUBTRACT Statement–Format 3–Corresponding

Format 3

```
SUBTRACT { CORRESPONDING } identifier-1 FROM identifier-2 [ ROUNDED ]
         { CORR          }

    [ ON SIZE ERROR imperative-statement ]
```

UNSTRING Statement

Format

```
UNSTRING identifier-1

    [ DELIMITED BY [ ALL ] { identifier-2 } [ , OR [ ALL ] { identifier-3 } ] . . . ]
                           { literal-1    }                { literal-2    }

    INTO identifier-4 [ , DELIMITER IN identifier-5 ] [ , COUNT IN identifier-6 ]

      [ , identifier-7 [ , DELIMITER IN identifier-8 ] [ , COUNT IN identifier-9 ] ] . . .

    [ WITH POINTER identifier-10 ] [ TALLYING IN identifier-11 ]

    [ ON OVERFLOW imperative-statement ]
```

USE Statement–EXCEPTION/ERROR Procedure–Format 1

Format

```
                                        { file-name-1 [ , file-name-2 ] . . . }
                                        { INPUT                               }
USE AFTER STANDARD { EXCEPTION } PROCEDURE ON { OUTPUT                        } .
                  { ERROR     }              { I-O                           }
                                        { EXTEND                              }
```

USE Statement–EXCEPTION/ERROR Procedure (TRANSACTION)–Format 2

Format

```
USE AFTER STANDARD { ERROR     }
                   { EXCEPTION }

    PROCEDURE ON { file-name-1 [ , file-name-2 ] . . . }
                 { I-O                                 } .
```

USE Statement–FOR DEBUGGING

Format

```
                   { [ ALL REFERENCES OF ] identifier-1 }
USE FOR DEBUGGING ON { file-name-1                      }
                   { procedure-name-1                   }
                   { ALL PROCEDURES                     }

    [ [ ALL REFERENCES OF ] identifier-2 ]
    [ file-name-2                        ] . . .
    [ procedure-name-2                   ]
    [ ALL PROCEDURES                     ]
```

WRITE Statement–Format 1–Sequential Files

Format 1

```
WRITE record-name [ FROM identifier-1 ]

    [                                                    ]
    [                     { { identifer-2 } { LINE  } }  ]
    [                     { { integer     } { LINES } }  ]
    [ { BEFORE } ADVANCING {                             } ]
    [ { AFTER  }           { { mnemonic-name }           } ]
    [                      { { PAGE          }           } ]
    [                                                    ]

    [ AT { END-OF-PAGE } imperative-statement ]
    [    { EOP         }                      ]
```

WRITE Statement–Format 2–Indexed and Relative Files

Format 2

```
WRITE record-name [ FROM identifier-1 ]

  [ FORMAT IS { identifier-2 } ]
              { literal-1    }

  [ INVALID KEY imperative-statement ]
```

WRITE Statement–Format 3–FORMATFILE Files

Format 3

```
WRITE record-name [ FROM identifier-1 ]

  [ FORMAT IS { identifier-2 } ]
              { literal-1    }

    [ { INDICATOR  [ IS  ] }                  ]
    [ { INDICATORS [ ARE ] } identifier-3     ]
    [ { INDIC              }                  ]

    [ AT { END-OF-PAGE } imperative-statement ]
    [    { EOP         }                      ]
```

WRITE Statement–Format 4–TRANSACTION File (Nonsubfile)

Format 4

```
WRITE record-name [ FROM identifer-1 ]

        FORMAT IS { identifier-2 }
                  { literal-1    }

        [ TERMINAL IS { identifier-3 } ]
                      { literal-2    }

        [ STARTING AT LINE { identifier-4 } ]
                           { literal-3    }

[ { BEFORE } ROLLING [ LINES ] { identifier-5 }                  ]
  { AFTER  }         [ LINE  ] { literal-4    }

          [ THROUGH ] { identifier-6 } { UP   }
          [ THRU    ] { literal-5    } { DOWN }

          { identifier-7 } [ LINES ]
          { literal-6    } [ LINE  ]                             ]

        [ { INDICATOR  [ IS  ] }                                 ]
        [ { INDICATORS [ ARE ] } identifier-8                    ]
        [ { INDIC              }                                 ]
```

WRITE Statement–Format 5–TRANSACTION File (Subfile)

Format 5

```
WRITE SUBFILE record-name [ FROM identifer-1 ]

    FORMAT IS { identifier-2 }
              { literal-1    }

    [ TERMINAL IS { identifier-3 } ]
                  { literal-2    }

    [ { INDICATOR  [ IS  ] }                  ]
    [ { INDICATORS [ ARE ] } identifier-4     ]
    [ { INDIC              }                  ]

    [ INVALID KEY imperative-statement ]
```

Chapter 6. Codes and Formats Used When Writing COBOL Programs

Conditional Expressions

Class Condition

Format

```
identifier IS [ NOT ] { NUMERIC    }
                      { ALPHABETIC }
```

Condition-Name Condition

Format

```
condition-name
```

Relation Condition

Format

```
                      { GREATER THAN }
                      { LESS THAN    }
operand-1 IS [ NOT ] { EQUAL TO     } operand-2
                      { >            }
                      { <            }
                      { =            }
```

© Copyright IBM Corp. 1994

Sign Condition

Format

```
                    { POSITIVE }
operand IS [ NOT ] { NEGATIVE }
                    { ZERO     }
```

Switch-Status Condition

Format

```
condition-name
```

Combined Condition

Format

```
condition { { AND } condition } . . .
          { { OR  }           }
```

Negated Simple Condition

Format

```
NOT simple-condition
```

Abbreviated Combined Relation Condition

Format

```
                    {                    [                 ]          }
                    {                    | GREATER THAN     |          }
                    {                    | LESS THAN        |          }
relation-condition { { AND } [ NOT ]     | EQUAL TO         | object } . . .
                  { { OR  }              | >                |          }
                    {                    | <                |          }
                    {                    | =                |          }
                    {                    [                 ]          }
```

Qualification of Data Reference Formats

Data Item Reference

Format

```
{ data-name-1    } [ { OF } data-name-2 ] . . .
{ condition-name }   { IN }
```

Procedure-Name Reference

Format

```
paragraph-name [ { OF } section-name ] . . .
               { IN }
```

COPY Library Reference

Format

```
text-name [ { OF } library-name ] . . .
          [ { IN }              ]
```

Note: `library-name` has the following format:

```
file name [ -library name ]
```

Subscripting

Format

```
{ data-name-1    } [ { OF } data-name-2 ] . . . ( subscript-1 [ , subscript-2 [ , subscript-3 ] ] )
{ condition-name } [ { IN }             ]
```

Indexing

Format

```
{ data-name-1    } [ { OF } data-name-2 ] . . . ( { index-name-1 [ {  } literal-2 ] }
{ condition-name } [ { IN }             ]       { literal-1                       }

    [ , { index-name-2 [ {  } literal-4 ] } ]  [  { index-name-3 [ {  } literal-6 ]  }  ] ] )
    [   { literal-3                       } ]  [ , { literal-5                        }  ] ]
```

COPY Statement

COPY Statement–Format 1–All Divisions

Format

```
COPY text-name [ { OF } file name [ -library name ] ]
               [ { IN }                             ]

     [          {                                    }       ]
     [          {   { ==pseudo-text-1== }    { ==pseudo-text-2== } }       ]
     [ REPLACING { , { identifier-1      } BY { identifier-2      } } . . . ] .
     [          {   { literal-1         }    { literal-2         } }       ]
     [          {   { word-1            }    { word-2            } }       ]
     [          {                                    }       ]
```

COPY Statement–Format 2–DDS Data Division

Format

```
     { DD-format-name  } [ -I     ] [ -INDICATOR  ]
COPY { DD-ALL-FORMATS  } [ -O     ] [ -INDICATORS ]
     { DDS-format-name } [ -I-O   ] [ -INDIC      ]
     { DDS-ALL-FORMATS }

     { OF } file name [ -library name ]
     { IN }

 [          {                                    }       ]
 [          {   { ==pseudo-text-1== }    { ==pseudo-text-2== } }       ]
 [ REPLACING { , { identifier-1      } BY { identifier-2      } } . . . ] .
 [          {   { literal-1         }    { literal-2         } }       ]
 [          {   { word-1            }    { word-2            } }       ]
 [          {                                    }       ]
```

Symbols Allowed in the PICTURE Clause

Symbol	Meaning
A	Alphabetic character or space
B	Space insertion character
P	Decimal scaling position (not counted in size of data item)
S	Operational sign (not counted in size of data item unless a SIGN clause with optional SEPARATE CHARACTER phrase is specified)
V	Assumed decimal point (not counted in size of data item)
X	Alphanumeric character (any from the EBCDIC set)
Z	Zero suppression character
9	Numeric character
1	Boolean character
0	Zero insertion character
/	Slash insertion character
,	Comma insertion character
+	Plus sign insertion edition control character
–	Minus sign editing control character
CR	Credit editing control character
DB	Debit editing control character
*	Check protect insertion character
'cs'	Currency sign insertion character (default is $)

Note: For a given program, the functions of the period and comma are exchanged if the clause `DECIMAL-POINT IS COMMA` is stated in the `SPECIAL-NAMES` paragraph. In this exchange, the rules for the period apply to the comma, and the rules for the commas apply to the period wherever they appear in the `PICTURE` clause.

Assignment-Names in the ASSIGN Clause

```
device [ -file name [ -attribute ] ]
```

where `attribute` can be one of the following:

- hopper [-association]
- `SI` (for separate indicator only)

file name:

- 1 to 10-character system-name

attribute:

- hopper [-association]
- SI (for separate indicator only)

hopper:

- P or S (for card devices only)

association:

- 0-9 (for card devices only)

Note: Card devices are not supported by System/38-Compatible COBOL, even though the devices are accepted by the syntax checker.

Table 1. ASSIGN Clause Devices

Device	Name and Default	Hopper	Association
READER	O QCARD96	O	O
PUNCH	O QCARD96	O	O
PUNCHFILE	O QCARD96	O	O
PRINT	O QCARD96	O	O
PRINTER	O QPRINT	N	N
TAPEFILE	O QTAPE	N	N
DISKETTE	O QDKT	N	N
DISK	R	N	N
DATABASE	R	N	N
WORKSTATION	R	N	N
FORMATFILE	R	N	N

R = Required; O = Optional; N = Not allowed.

Function-Names in the SPECIAL-NAMES Paragraph

function-name-1

CONSOLE	System-console
SYSTEM-CONSOLE	System-console
REQUESTOR	Work station or batch input stream
CSP	Suppress spacing after printing a line
C01	Skip to next page
OPEN-FEEDBACK	Information about a file
I-O-FEEDBACK	Information about the last I-O operation
ATTRIBUTE-DATA	Information about a work station or communications device
LOCAL-DATA	Data area created by the system for every job

function-name-2

UPSI-0 through UPSI-7	Program switches associated with condition-names
SYSTEM-SHUTDOWN	Internal switches associated with condition-names

Figurative Constants

The following figurative constants can be used:

```
ALL "literal"
HIGH-VALUE
HIGH-VALUES
LOW-VALUE
LOW-VALUES
QUOTE
QUOTES
SPACE
SPACES
ZERO
ZEROES
ZEROS
```

Status Key Values and Meanings

Table 2 (Page 1 of 7). Status Key Values and Meanings

Status Key 1	Status Key 2	Meaning	When Set (OS/400 Exceptions Monitored, Condition Detected)
0		**Successful Completion**	
	0	Successful Completion	No error condition occurred during the I-O operation.
1		**At End of File**	
	0	End of file	CPF4740, CPF5001, CPF5025.
	2	No modified subfile record found (IBM extension)	CPF5037.

Table 2 (Page 2 of 7). Status Key Values and Meanings

Status Key 1	Status Key 2	Meaning	When Set (OS/400 Exceptions Monitored, Condition Detected)
2		**Invalid Key**	
	1	Sequence error	REWRITE to an indexed file with sequential access and key for REWRITE ≠ key from previous READ, or WRITE to an indexed file with sequential access and key values for succeeding writes are not in ascending sequence.
	2	Duplicate key when duplicates are not allowed	CPF4759, CPF5008, CPF5026, CPF5034, CPF5084, CPF5085, or WRITE to an indexed file with sequential access and key values for succeeding writes are not in ascending sequence.
	3	No record found	CPF5001, CPF5006, CPF5013, CPF5020, CPF5025.
	4	Boundary violation	CPF5006, CPF5018, CPF5021, CPF5043, CPF5272, if organization is not sequential.
3		**Permanent Error**	
	0	Permanent Error	CPF4192, CPF5030, CPF5101, CPF5102, CPF5129, CPF5143.
	4	Boundary violation	CPF5018, CPF5116, CPF5272, if organization is sequential.

Table 2 (Page 3 of 7). Status Key Values and Meanings

Status Key 1	Status Key 2	Meaning	When Set (OS/400 Exceptions Monitored, Condition Detected)
9		**Other Errors**	
	0	Other errors:	
		•File not found	CPF4101 if a USE is applicable for the file.
		•Member not found	CPF4102 if a USE is applicable for the file.
		•Level check error	CPF4131.
		•Unexpected I-O exceptions	The following exceptions are monitored generically: CPF4101 through CPF4399 CPF4501 through CPF4699 CPF4701 through CPF4899 CPF5001 through CPF5099 CPF5101 through CPF5399 CPF5500 through CPF5699 These exceptions are caught and FILE STATUS is set to 90. If a USE procedure is applicable, it is processed. Otherwise, the program ends and gives the operator the exception and the option to cancel, take a partial dump, or take a full dump.

Table 2 (Page 4 of 7). Status Key Values and Meanings

Status Key 1	Status Key 2	Meaning	When Set (OS/400 Exceptions Monitored, Condition Detected)
9		**Other Errors (continued)**	
	1	Undefined or unauthorized access type	CPF2207, CPF4104, CPF4236, CPF5057, CPF5109, CPF5134, CPF5279.
	2	Logic error: •File locked •File already open •I-O to closed file •READ after end of file •CLOSE on unopened file	CPF4102, CPF4106, CPF4132, CPF4194, CPF4740, CPF5013, CPF5067, CPF5070, CPF5119, CPF5132, CPF5145, CPF5146, CPF5149, CPF5176, CPF5183, CPF5209.
	4	No current record pointer	REWRITE/DELETE with sequential access, and last operation was not a successful READ.

Table 2 (Page 5 of 7). Status Key Values and Meanings

Status Key 1	Status Key 2	Meaning	When Set (OS/400 Exceptions Monitored, Condition Detected)
9		**Other Errors (continued)**	
	5	Invalid or incomplete file information	(1) Duplicate keys specified in COBOL program, but indexed data base file created with unique key; or (2) Duplicate keys not specified in COBOL program, and indexed data base file created allowing duplicate keys.
	A	Job has been cancelled in a controlled manner by CL command CNLJOB, TRMSBS, TRMCPF, or PWRDWNSYS	CPF4741. Escape message sent during a READ from invited program device (multiple device files only).
	D	Record is locked	CPF5027, CPF5032.

Table 2 (Page 6 of 7). Status Key Values and Meanings

Status Key 1	Status Key 2	Meaning	When Set (OS/400 Exceptions Monitored, Condition Detected)
9		**Other Errors (continued)**	
	H	ACQUIRE operation failed	Resource owned by another program, or unavailable. (9H is the result when an ACQUIRE operation causes any of the OS/400 exceptions monitored for 90 or 9N to occur).
	I	WRITE operation failed	CPF4702, CPF4737, CPF5052, CPF5076.
	K	Invalid format-name; format not found	CPF5022, CPF5023, CPF5053, CPF5054, CPF5121, CPF5152, CPF5153, CPF5186, CPF5187, CPF5003.
	M	Last record written to subfile	

Table 2 (Page 7 of 7). Status Key Values and Meanings

Status Key 1	Status Key 2	Meaning	When Set (OS/400 Exceptions Monitored, Condition Detected)
9		**Other Errors (continued)**	
	N	Temporary (potentially recoverable) hardware I-O error	CPF4145, CPF4146, CPF4193, CPF4229, CPF4291, CPF4299, CPF4354, CPF4526, CPF4542, CPF4577, CPF4592, CPF4602, CPF4603, CPF4611, CPF4612, CPF4616, CPF4617, CPF4622, CPF4623, CPF4624, CPF4625, CPF4628, CPF4629, CPF4630, CPF4631, CPF4632, CPF4705, CPF5107, CPF5128, CPF5166, CPF5198, CPF5280, CPF5282, CPF5287, CPF5293, CPF5352, CPF5353, CPF5517, CPF5524, CPF5529, CPF5530, CPF5532, CPF5533.
	P	OPEN failed because file cannot be placed under commitment control	CPF4285, CPF4293, CPF4326, CPF4327, CPF4328, CPF4329,

Communicating Your Comments to IBM

Application System/400
System/38-Compatible COBOL
Reference Summary

Publication No. SX09-1286-00

If there is something you like—or dislike—about this book, please let us know. You can use one of the methods listed below to send your comments to IBM. If you want a reply, include your name, address, and telephone number. If you are communicating electronically, include the book title, publication number, page number, or topic you are commenting on.

The comments you send should only pertain to the information in this book and its presentation. To request additional publications or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give it to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - United States and Canada: 416-448-6161
  - Other countries: (+1)-416-448-6161
- If you prefer to send comments electronically, use the network ID listed below. Be sure to include your entire network address if you wish a reply.
  - Internet: torrcf@vnet.ibm.com
  - IBMLink: toribm(torrcf)
  - IBM/PROFS: torolab4(torrcf)
  - IBMMAIL: ibmmail(caibmwt9)

Readers' Comments — We'd Like to Hear from You

Application System/400
System/38-Compatible COBOL
Reference Summary

Publication No. SX09-1286-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____ _____

Phone No. _____ _____

PLACE
POSTAGE
STAMP
HERE

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 EGLINTON AVENUE EAST
NORTH YORK ONTARIO CANADA M3C 1H7

Fold and Tape **Please do not staple** Fold and Tape

Readers' Comments — We'd Like to Hear from You
SX09-1286-00

IBM®

IBM®

Program Number: 5763-CB1

Printed in U.S.A.

SX09-1286-00